UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/675,267	09/29/2000	Ivan Chow	Sprint IDF 1421 (4000-012	1016

7590 12/18/2003

Steven J Funk
6450 SPRINT PARKWAY
MS: KSOPHN0312 3A371
OVERLAND PARK, KS 66251

EXAMINER	
ALI, SYED J	

ART UNIT	PAPER NUMBER
2127	

DATE MAILED: 12/18/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/675,267	09/29/2000	Ivan Chow	Sprint IDF 1421 (4000-012	1016

7590 11/10/2003

Steven J Funk
Sprint Law Department
8140 Ward Parkway
Kansas City, MO 64114

EXAMINER	
ALI, SYED J	
ART UNIT	PAPER NUMBER
2127	3

DATE MAILED: 11/10/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/675,267	**Applicant(s)** CHOW ET AL.
	Examiner Syed J Ali	**Art Unit** 2127

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 September 2000* .

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-15* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)□ The proposed drawing correction filed on _____ is: a)□ approved b)□ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)□ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____ .

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) □ The translation of the foreign language provisional application has been received.

15)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 2 .
4) □ Interview Summary (PTO-413) Paper No(s). _____ .
5) □ Notice of Informal Patent Application (PTO-152)
6) □ Other: .

DETAILED ACTION

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1-13 are rejected under 35 U.S.C. 102(e) as being anticipated by Hollberg et al. (USPN 6,356,955) (hereinafter Hollberg).

As per claim 1, Hollberg discloses a method for a client application in a single-threaded environment controlled by a user to request and receive multiple messages asynchronously from a destination application, the method comprising:

the client application sending a first request to a software agent running in a single-threaded environment (col. 12 lines 5-12, "For asynchronous requests, a request object is allocated by the application and passed to the OOI"; col. 13 lines 45-52, "The OOI is single threaded");

the client application sending a second request to a software agent prior to receiving a response to the first request from the software agent (col. 12 lines 15-32, "Since several proxy agent objects may exist at the same time in the same application, several wait methods are available");

the client application continuing on in execution in its single-threaded environment prior to receiving responses to the first request or the second request from the software agent (col. 8 lines 60-61, "Asynchronous methods return control to the application as soon as a CMIP request is sent");

the software agent registering the first request and forwarding the first request to the destination application (col. 7 lines 44-53, "Each PMO C++ class provides a set of object class specific methods through which a management application can conveniently submit CMIS requests to query or manipulate the real managed object in the agent");

the software agent beginning to cyclically poll the destination application for a first response to the first request (col. 11 lines 27-30, "poll( ) checks with XMP whether something has arrived");

the software agent in between polling cycles registering the second request and forwarding the second request to the destination application (col. 9 lines 4-7, "Incoming notifications are queued in the event queue 35 of the responsible proxy agent 30. The application can thus process notification at its leisure", wherein the proxy agent can handle the request at any time that it is not executing another method, such as the aforementioned poll( ) method);

the software agent beginning to cyclically poll the destination application for a second response to the second request, wherein such polling cycles in sequence with the polling for the first response to the first request (col. 11 lines 45-53, "Notifications are received at any time when the proxy agent 30 is receiving messages from its XMP session", wherein the poll( ) method is run periodically to check for incoming messages, and since the OOI is single-threaded, the polling cycles inherently must be in sequence with each other such that they can be scheduled properly);

the destination application generating the first response to the first request and forwarding the first response to the software agent in response to polling from the software agent (col. 13 line 66 - col. 14 line 11, "Incoming responses for pending requests are represented by objects which have been derived from the ASN.1 definition of CMIP. These objects are put into the reply queue of the request object");

the software agent receiving the first response from the destination application, ceasing cyclically polling the destination application for the first response, and storing the first response associated with the first request, wherein such actions of receiving, ceasing, and storing occur in between the continuing polling cycles (col. 13 line 66 - col. 14 line 11, "These [incoming response] objects are put into the event queue of their proxy agent");

the destination application generating the second response to the second request and forwarding the second response to the software agent in response to polling from the software agent (col. 13 line 66 - col. 14 line 11, wherein it is shown above how Hollberg supports the use of several pending requests, and each incoming response is handled in the same way);

the software agent receiving the second response from the destination application, ceasing cyclically polling the destination application for the second response, and storing the second response associated with the second request, wherein such actions of receiving, ceasing, and storing occur in between the continuing polling cycles (col. 13 line 66 - col. 14 line 11, wherein it is shown above how Hollberg supports the use of several pending requests, and each incoming response is handled in the same way);

the client application polling the software agent for the first response to the first request and the software agent responding by forwarding the first response from storage to the client application and deleting the first response from storage, wherein the actions of responding by forwarding and deleting occur in between the continuing polling cycles (col. 13 lines 44-52, "To facilitate the processing of arriving reply messages, the OOI offers the possibility of defining callback methods, which will be activated as soon as a reply message or a notification has been received. The OOI is single threaded, therefor callbacks are invoked only during wait( ) or poll( ) calls and not while the application is processing"; col. 13 lines 31-40, Request objects must be explicitly deleted by the application, even if the related proxy agent is deleted. All response queue elements included in the request object are automatically deleted with the request object");

the client application polling the software agent for the second response to the second request and the software agent responding by forwarding the second response from storage to the client application and deleting the second response from storage, wherein the actions of responding by forwarding and deleting occur in between the continuing polling cycles (col. 13 line 66 - col. 14 line 11, wherein it is shown above how Hollberg supports the use of several pending requests, and each incoming response is handled in the same way).

As per claim 2, Hollberg discloses the method of claim 1, wherein the action of the client application of sending the second request occurs after the actions of the software agent of registering the first request and forwarding the first request to the destination application and of beginning to cyclically poll the destination application for a first response to the first request (col. 11 line 45 - col. 12 line 12, "The application explicitly passes control to the OOI run-time system [RTS] by invoking a method to check upon or to wait for the reception of incoming messages", wherein the application sends requests to the OOI at any time, and the proxy agent processes and forwards the request during free processing cycles).

As per claim 3, Hollberg discloses the method of claim 1, wherein the action of the client application of sending the second request occurs before the actions of the software agent of registering the first request and forwarding the first request to the destination application and of beginning to cyclically poll the destination application for a first response to the first request (col. 11 line 45 - col. 12 line 12, "The application explicitly passes control to the OOI run-time system [RTS] by invoking a method to check upon or to wait for the reception of incoming messages", wherein the application sends requests to the OOI at any time, and the proxy agent processes and forwards the request during free processing cycles).

As per claim 4, Hollberg discloses the method of claim 1, wherein the action of the destination application of generating the second response to the second request occurs prior to the action of the destination application of generating the first response to the first request (col. 11 line 45 - col. 12 line 12, "Notifications are received at any time when the proxy agent 30 is receiving messages from its XMP session. The management application may process the notifications in the queue 35 at any time"); and

wherein the action of the software agent of receiving the second response from the destination application, and performing the actions of ceasing polling and storing related to the second response occur prior to the actions of the software agent of receiving the first response from the destination application, and performing the actions of ceasing polling and storing related to the first response (col. 11 line 45 - col. 12 line 12, "Notifications are received at any time when the proxy agent 30 is receiving messages from its XMP session. The management application may process the notifications in the queue 35 at any time", wherein the actions of ceasing polling and storing the response are a part of the procedure of processing the notifications).

As per claim 5, Hollberg discloses the method of claim 1, wherein the action of the destination application of generating the second response to the second request occurs after the action of the destination application of generating the first response to the first request (col. 11 line 45 - col. 12 line 12, "Notifications are received at any time when the proxy agent 30 is receiving messages from its XMP session. The management application may process the notifications in the queue 35 at any time"); and

wherein the action of the software agent of receiving the second response from the destination application, and performing the actions of ceasing polling and storing related to the second response occur after the actions of the software agent of receiving the first response from the destination application, and performing the actions of ceasing polling and storing related to the first response (col. 11 line 45 - col. 12 line 12, "Notifications are received at any time when the proxy agent 30 is receiving messages from its XMP session. The management application may process the notifications in the queue 35 at any time", wherein the actions of ceasing polling and storing the response are a part of the procedure of processing the notifications).

As per claim 6, Hollberg discloses the method of claim 4, wherein the action of the client application polling the software agent for the second response to the second request and the subsequent forwarding and deleting actions related to the second response to the second request occur prior to the action of the client application polling the software agent for the first response to the first request and the subsequent forwarding and deleing action related to the first response to the first request (col. 12 lines 14-32, "Since several proxy agent objects may exist at the same time in the same application, several wait methods are available: The global wait method returns if anything was received on any proxy agent object, [i.e. on any XMP session]. The wait method of the proxy agent returns if anything was received on that session. The wait method of the request object returns if anything was received on that request object", wherein the action of the response being forwarded to the client is done in response to a response being received, and since several proxies are active at one time, and is not necessarily related to the order in which the requests were issued, but rather, in the order responses are received).

As per claim 7, Hollberg discloses the method of claim 4, wherein the action of the client application polling the software agent for the second response to the second request and the subsequent forwarding and deleting actions related to the second response to the second request occur after the action of the client application polling the software agent for the first response to the first request and the subsequent forwarding and deleting actions related to the first response to the first request (col. 12 lines 14-32, "Since several proxy agent objects may exist at the same time in the same application, several wait methods are available: The global wait method returns if anything was received on any proxy agent object, [i.e. on any XMP session]. The wait method of the proxy agent returns if anything was received on that session. The wait method of the request object returns if anything was received on that request object", wherein the action of the response being forwarded to the client is done in response to a response being received, and since several proxies are active at one time, and is not necessarily related to the order in which the requests were issued, but rather, in the order responses are received).

As per claim 8, Hollberg discloses the method of claim 1, wherein the action of the client application polling the software agent for the first response to the first request and the subsequent forwarding and deleting actions related to the first response to the first request occur prior to the actions of the destination application generating the second response to the second request and forwarding the second response to the software agent and prior to subsequent actions of the software agent and of the client application involving the second response (col. 12 lines 14-32, "Since several proxy agent objects may exist at the same time in the same application, several wait methods are available: The global wait method returns if anything was received on any proxy agent object, [i.e. on any XMP session]. The wait method of the proxy agent returns if anything was received on that session. The wait method of the request object returns if anything was received on that request object", wherein the action of the response being forwarded to the client is done in response to a response being received, and since several proxies are active at one time, and is not necessarily related to the order in which the requests were issued, but rather, in the order responses are received).

As per claim 9, Hollberg discloses the method of claim 1, wherein the destination application is resident on a server remote to the client application and to the software agent (Fig. 1, elements 80, wherein the remote managed objects are shown to be on a remote server, and connected to the application 10 through a communication infrastructure 50); and

wherein the actions of the software agent of storing the responses received from the destination application stores such responses on a server local to the software agent (Fig. 1 elements 35, wherein the local event queues 35 are where responses are stored while awaiting processing, and are local to the proxy agent, i.e., software agent, 30); and

wherein the actions of the software agent of responding by forwarding the responses from storage to the client application forwards such responses from storage on a server local to the software agent (Fig. 1, elements 35, arrow [3], wherein the responses are forwarded from the local event queue 35 to the application 10 through via communication arrow [3]).

As per claim 10, Hollberg discloses the method of claim 9, wherein the software agent is resident on a server local to the client application (Fig. 1 element 30, wherein the proxy agent, i.e. software agent, is local to the application 10); and

wherein the actions of the software agent of storing the responses received from the destination application stores such responses on a server local to the client application (Fig. 1, elements 35, wherein the responses to from the destination applications are stored in event queues 35 while awaiting processing); and

wherein the actions of the software agent of responding by forwarding the responses from storage to the client application forwards such responses from storage on a server local to the client application (Fig. 1, elements 10, 35, arrow [3], wherein the forwarding of responses goes from event queue 35 to application 10 via communication arrow [3], all of which occurs on the same machine).

As per claim 11, Hollberg discloses the method of claim 9, wherein the remote destination application comprises a destination server application (Fig. 1 element 80, wherein the managed object 80 is the remote application) and a destination client application which manages requests to and responses from the destination server application (Fig. 1 elements 60, 70, wherein the XMP service handles communication between the managed object and the proxy agent); and

wherein the actions of forwarding the requests to the destination application comprise forwarding the requests to the destination client application (Fig. 1, elements 70, 80, wherein the requests are forwarded to the managed objects via the agent kernel 70); and

wherein the actions of the software agent of cyclically polling the destination application comprise cyclically polling the destination client application for the responses to the request (col. 11 lines 27-32, "The following methods to wait for the completion of a request or a notification are provided:...poll( ) checks with XMP whether something has arrived"); and

wherein the actions of the destination application forwarding the responses to the software agent in response to polling from the software agent comprise the destination client application forwarding responses to the software agent (Fig. 1, wherein communication is shown that allows the response from the managed object to be passed forward to the proxy agent).

As per claim 12, Hollberg discloses the method of clam 1, further comprising:

the client application, in conjunction with sending a first request to the software agent, sending a command to register a callback associated with such first request (col. 13 line 44 - col. 14 line 32, “When using several asynchronous requests at the same time, replies may appear in any order. To facilitate the processing of arriving reply messages, the OOI offers the possibility of defining callback methods, which will be activated as soon as a reply message or a notification has been received”);

the software agent, in conjunction with the actions of receiving the first response from the destination application, ceasing cyclically polling the destination application for the first response, and storing the first response associated with the first request, further takes the action in response to the callback of notifying the user that the first response has been received (col. 13 line 44 - col. 14 line 32, “The callback informs the application that something has been received, and that the queue structures were updated”).

As per claim 13, Hollberg discloses the method of claim 12, wherein the command to register a callback comprises a command resulting in instantiation of a callback object associated with the software agent (col. 13 line 44 - col. 14 line 32, “Incoming responses for pending request are represented by objects which have been derived from the AND.1 definitions of CMIP. These objects are put into the reply queue of the request object”); and

wherein the action of the software agent of notifying the user comprises the callback object responding to the storage of the first response associated with the first request by notifying the user that the first response has been received (col. 13 line 44 - col. 14 line 32, “Incoming notifications are represented by objects which also have been derived from the ASN.1 definitions of CMIP. These objects are put into the event queue of their proxy agent. Then the callback method incomingNotificationCallback( ) which is defined by the application for the notification queue is executed with the notification object as parameter").

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Hollberg in view of Chang et al. (USPN 6,338,078) (hereinafter Chang) in view of Burns (USPN 6,098,090)

As per claim 14, Chang discloses the following limitations not shown by Hollberg, specifically, the method of claim 12, wherein the action of notifying the user that the first response has been received comprises sending a pop-up to the user notifying the user (col. 4 lines 12-23, "the user will view various objects such as a cursor and pop up or pop down menus on the display 21 which may be manipulated by means of various pointing devices").

It would have been obvious to one of ordinary skill in the art to combine Hollberg with Chang since the use of pop up dialogs allows the immediate notification to the user that an event has occurred. Particularly in the case where the event cannot be immediately processed, the pop up would allow the user to be aware of the fact that a response has been received and will be serviced shortly.

Burns discloses the following limitations not shown by the modified Hollberg, specifically that the notification of the user is done without interrupting the thread of execution of the client application (col. 4 line 63 - col. 5 line 21, "Because task 36 executes in the single background processor thread 19, task 36 executes from start to finish without interruption").

It would have been obvious to one of ordinary skill in the art to combine the modified Hollberg with Burns since in circumstances such as when the client application is executing a high priority task, it may be critical that the task finish completion before other processing is done. Specifically, it is well known to assign priorities to tasks within a system. If a determination were made that the currently executing task is of a higher importance than the task that is awaiting a response, it would be most beneficial to continue processing the current task to completion before servicing the response.

5. Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over Hollberg in view of Chang.

As per claim 15, Chang discloses prior art that teaches the following limitations not shown by Hollberg, specifically the method of claim 12, wherein the action of notifying the user that the first response has been received comprises notifying the client application that the first response has been received and the client application interrupting its thread of execution to notify the user that the first response has been received in response to the notification from the software agent (col. 5 lines27-65, “incoming packets on the network 30 will each generate an interrupt. Conventionally, when the system of Fig. 2 thus receives an interrupt, the single CPU of the system is thus interrupted and will begin handling the incoming packets. When the CPU is through processing all packets, the interrupt handling has been completed, and the CPU returns to a normal context”).

It would have been obvious to one of ordinary skill in the art to combine Hollberg with the prior art shown by Chang since in cases where the incoming notification is in response to an urgent request, or simply of a higher priority, the processing of that response should occur as soon as possible. By interrupting the thread of execution to process the request could resolve data dependencies that other blocked requests may be waiting for, or other such circumstances that require the immediate processing of a reply. Thus, interrupting the thread such that the response is handled immediately is beneficial in the sense that urgent requests do not have to wait once a reply is available.

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The additional cited references pertain to various aspects of the claimed invention, such as a software agent, i.e., request broker, or various ways of forwarding requests and responses to and from a remote application.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William A Grant can be reached on (703) 308-1108. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Syed Ali
October 27, 2003

MAJID A. BANANKHAH
PRIMARY EXAMINER